Setting up a Simple Database Infrastructure using Amazon RDS

Introduction

The goal of this assignment was to create a SQL database using the AWS console and establish a connection to it from a local machine using MySQL Workbench. This involved setting up an RDS instance in AWS, selecting the appropriate configurations such as database engine and credentials, and ensuring that the database was accessible from our local environment. Through this process, we explored how cloud-hosted databases can be effectively managed and accessed using familiar local tools, making it easier to integrate cloud services into everyday development tasks.

Group Members

- Esther Acheampong
- Eugene Essuman
- Grace Wiredu
- Habiba Adam Salisu
- Benedicta Maayuomle Djangmah
- Bernard Solodzi
- Osman Abdul
- Michael Ohene Aboagye
- Nick Anderson
- Rachel Atia
- Vera Sekyere
- Azal Innocentia
- Akoto Nana Yaw Appiah
- Rhoda Kudor
- Benedicta Opoku-Amankwaah
- Jessica Azumah
- Rose Anyama Owusu
- Mohammed Razak Mikdar
- Clement Owusu Bempah

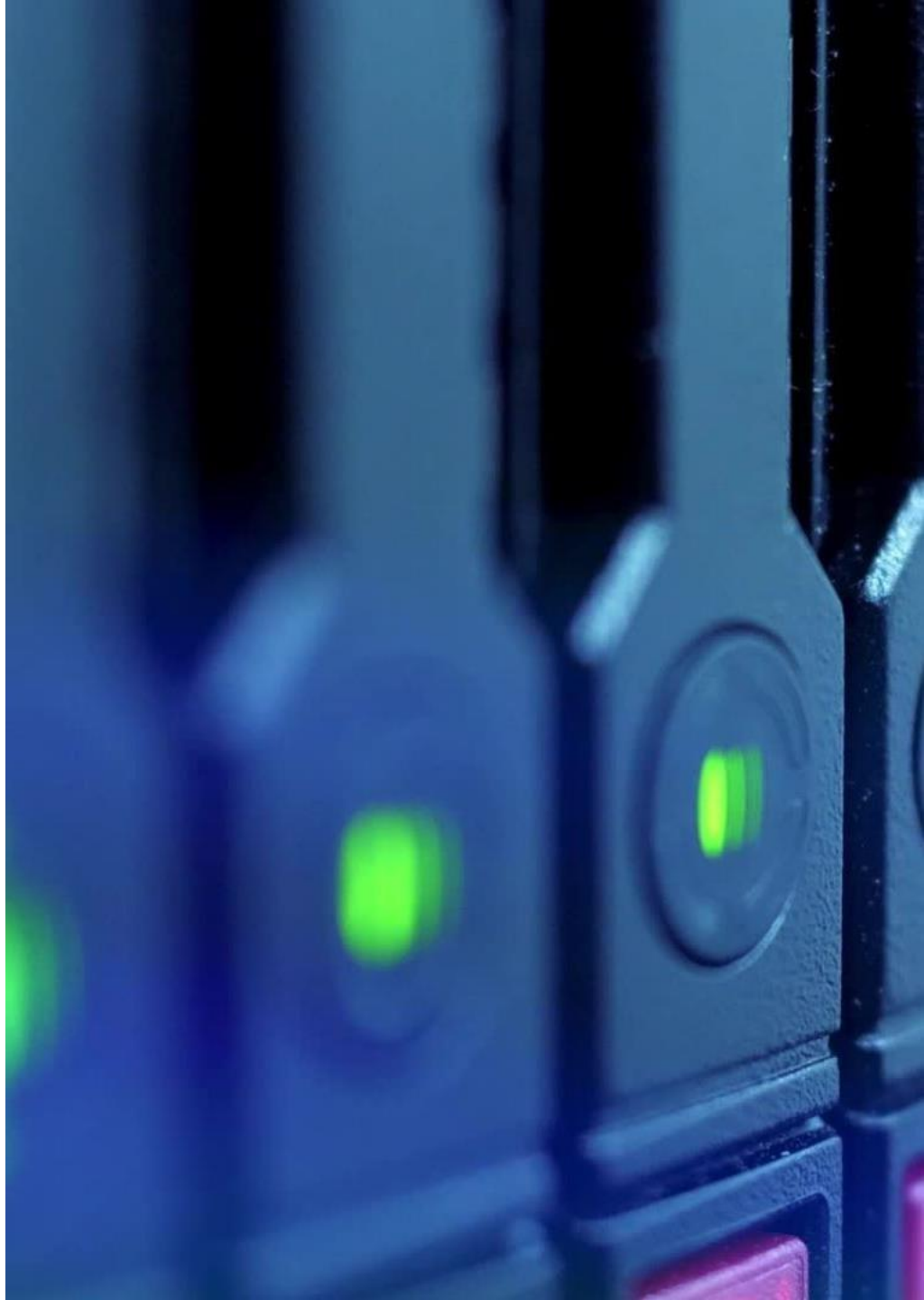
Step 1 to 4 – Launch RDS on AWS Console

Process:

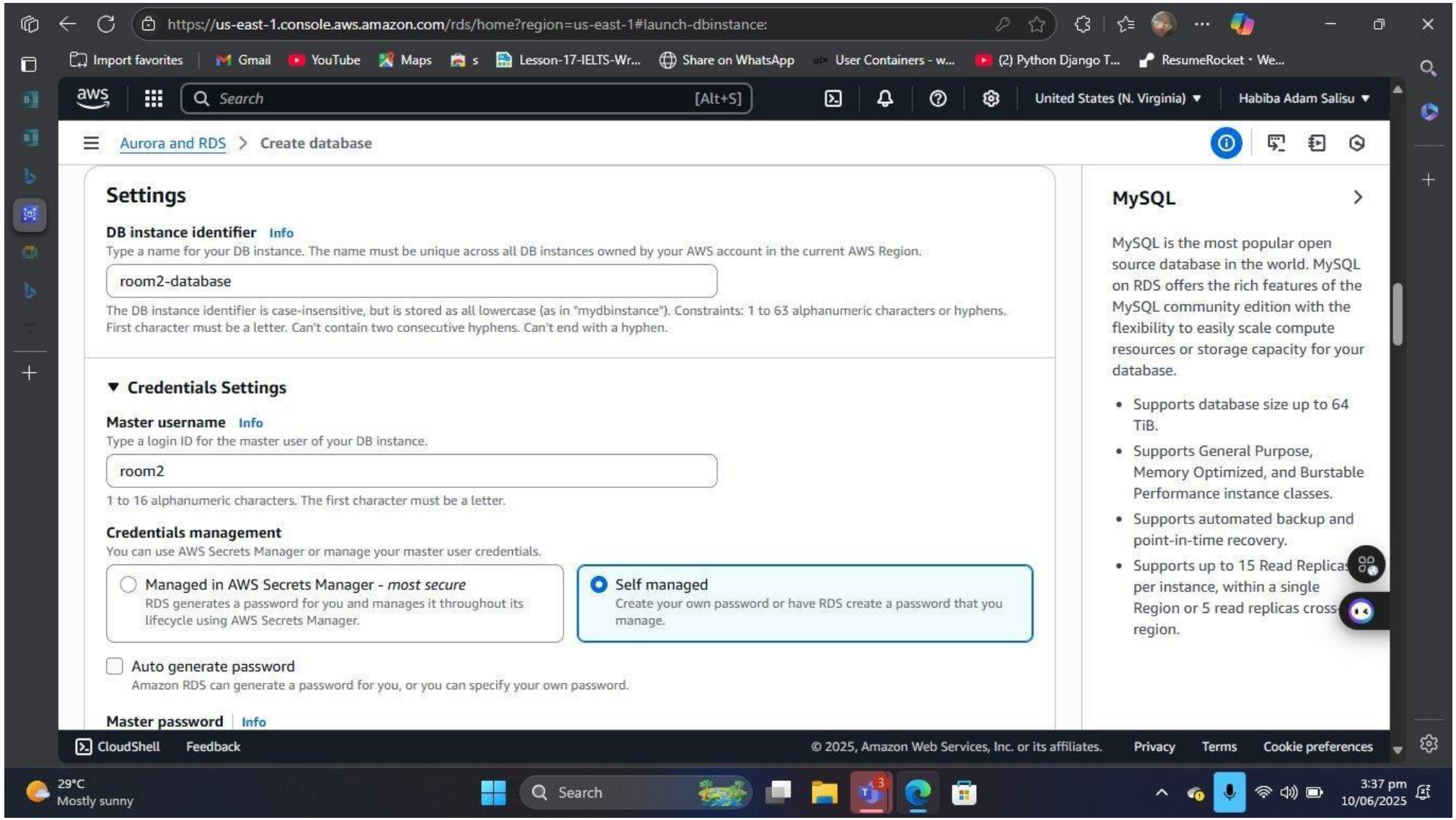
- Logged into AWS Management Console.
- Navigated to **Amazon RDS > Databases > Create Database > Selected A Region(Virginia)**

Lessons Learned:

- AWS RDS supports multiple database engines (MySQL, PostgreSQL, etc.).
- GUI-based deployment makes setup easy for beginners and professionals.



Step 4 to 5 - Creation of Database Instance named: room2-database



Step 6 and 7

DB Instance Settings: Provide a unique DB instance identifier, a master username, and a secure password (8–41 printable ASCII characters, excluding /, ", and @).

Instance Class: Choose db.t2.micro (1 vCPU, 1 GiB RAM) for basic usage.

Storage Configuration: Use **General Purpose (SSD)** with **20 GB allocated storage** (scalable up to 64 TB); autoscaling is not enabled in this tutorial.

Multi-AZ Deployment: Optional feature that adds high availability with a standby replica in another Availability Zone, but incurs additional cost.

See in Fig. 2

View of Step 6 and 7

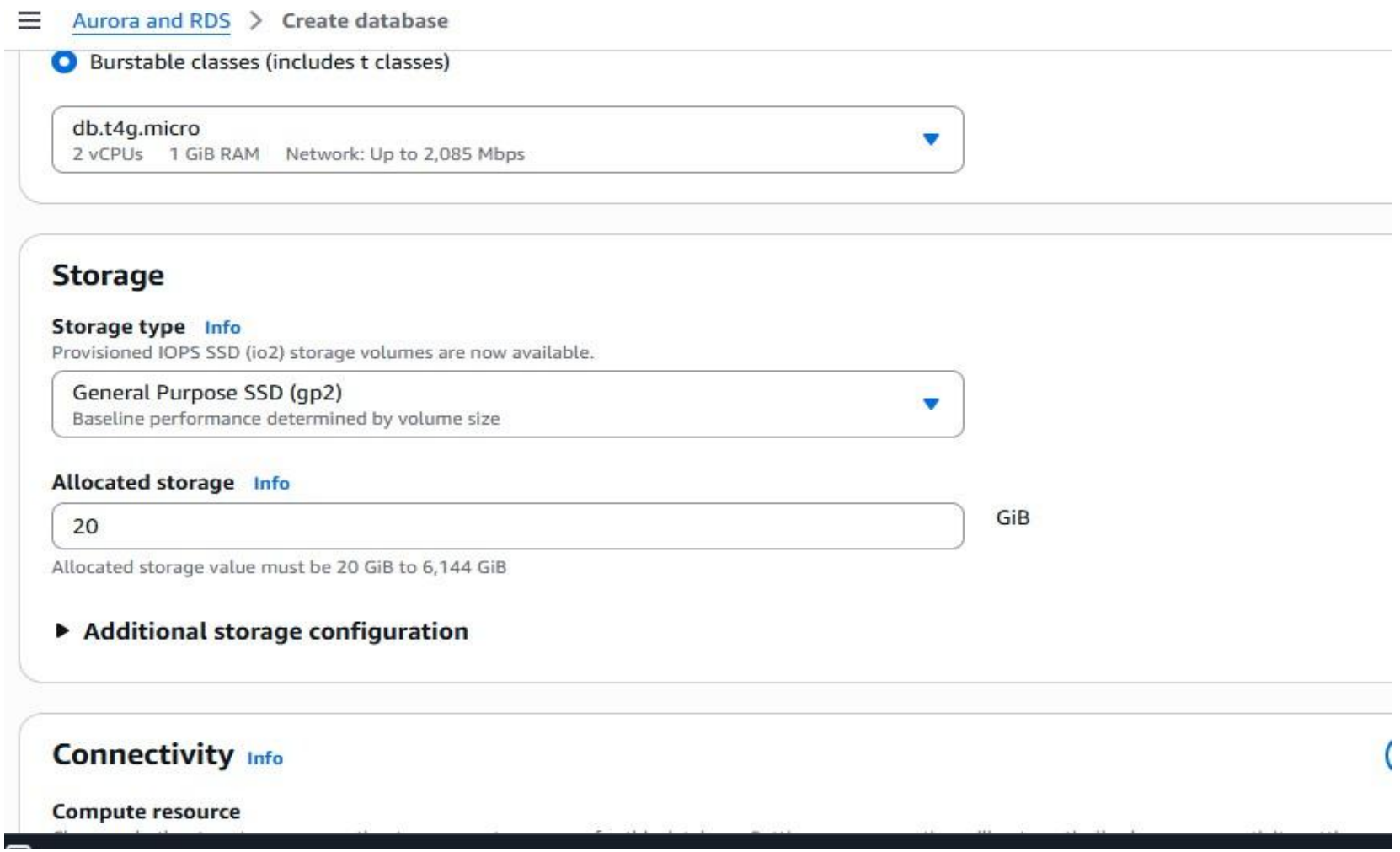
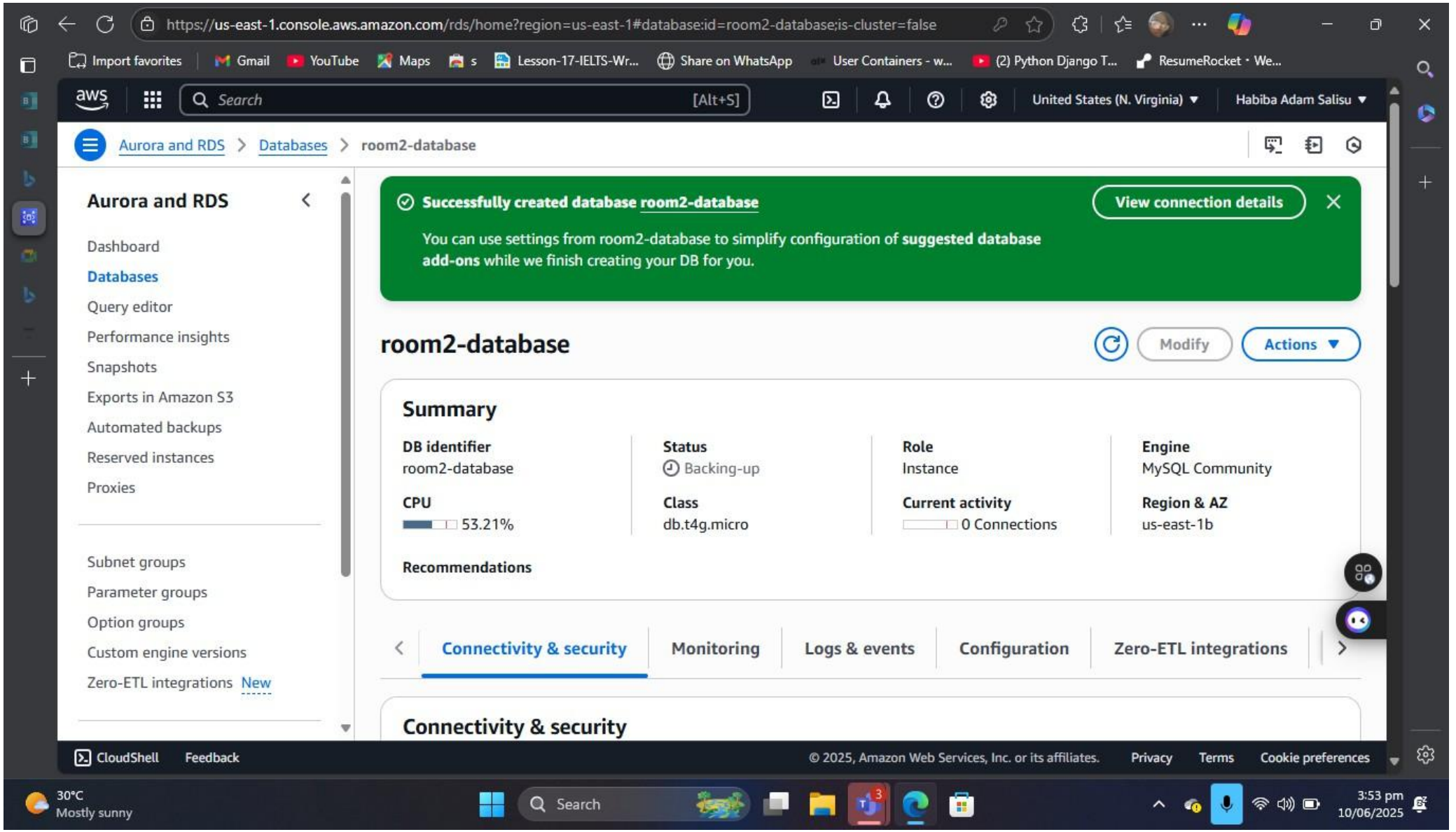
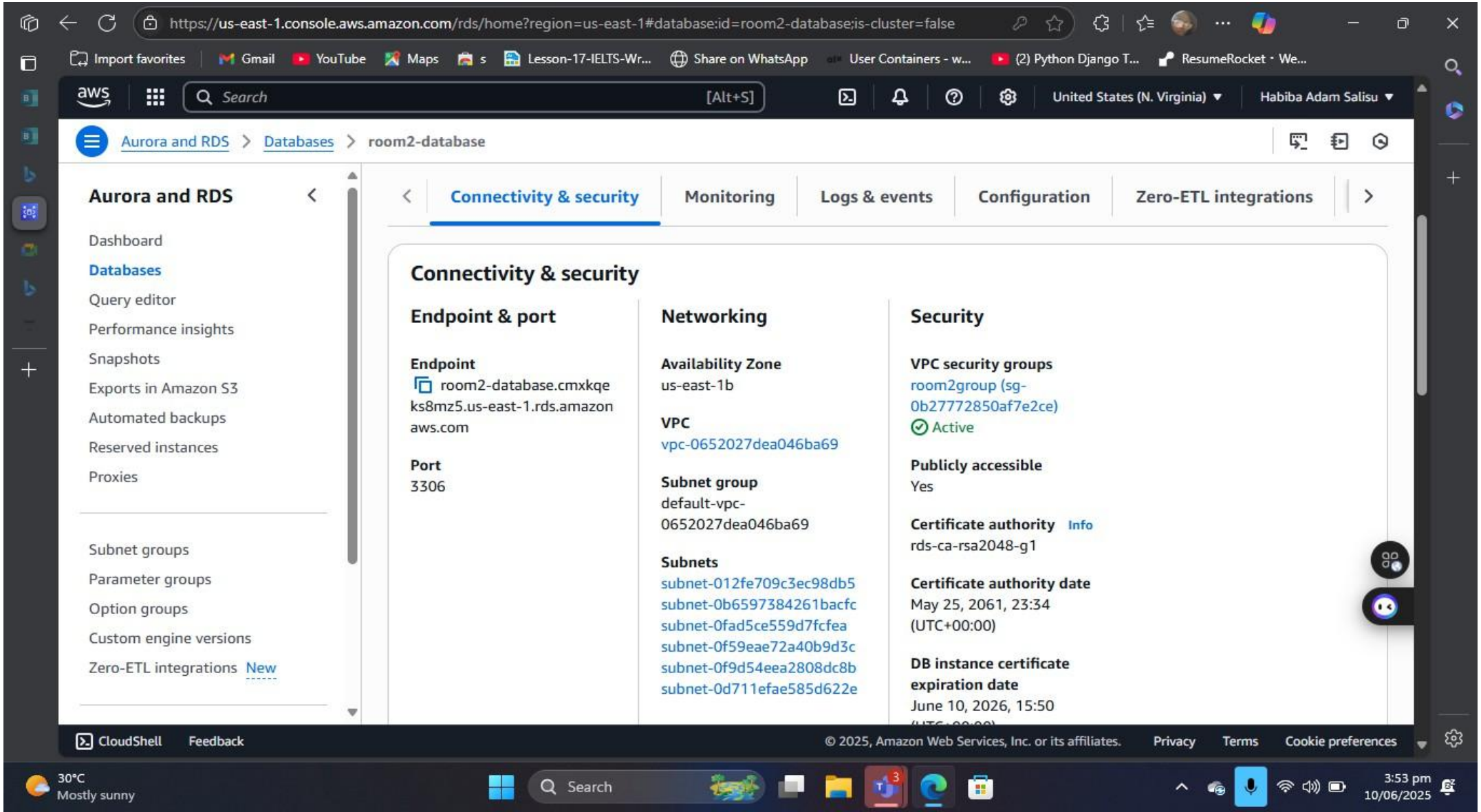


Fig. 2

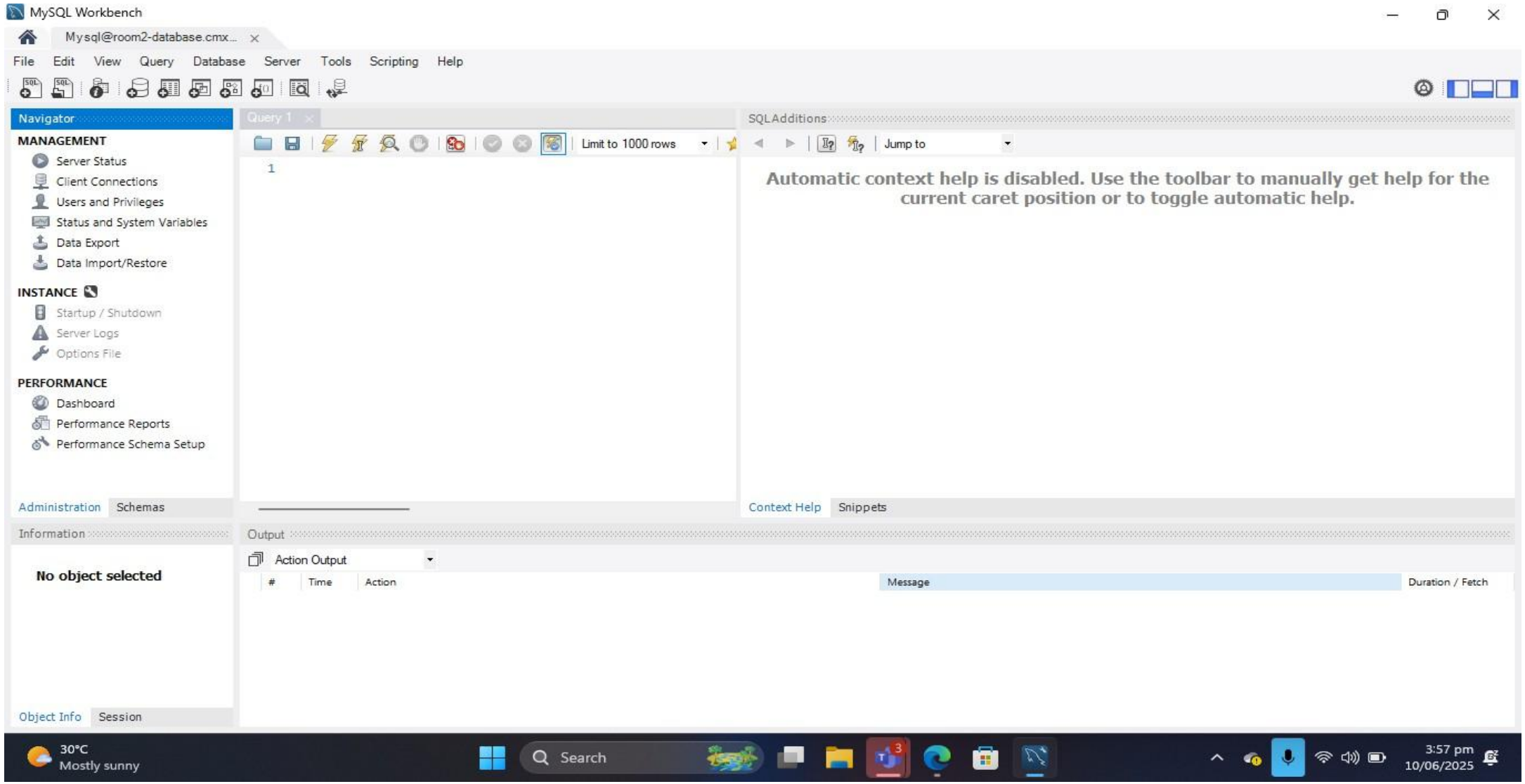
Details of Database: **room2-database** created



Copying of Necessary Details For Local Connection



Launch MySQL Workbench



Query For Creating A Table in MySQL Workbench

```
CREATE TABLE `employee` (
`id` int NOT NULL  AUTO_INCREMENT,
`name’   varchar(255) DEFAULT NULL,
`email` varchar(255) DEFAULT NULL,
`city` varchar(255) DEFAULT NULL,
PRIMARY KEY (`id`) )
```

We used the script above to create a table in MySQL workbench

See in Fig. 3

Creating A Table With Details

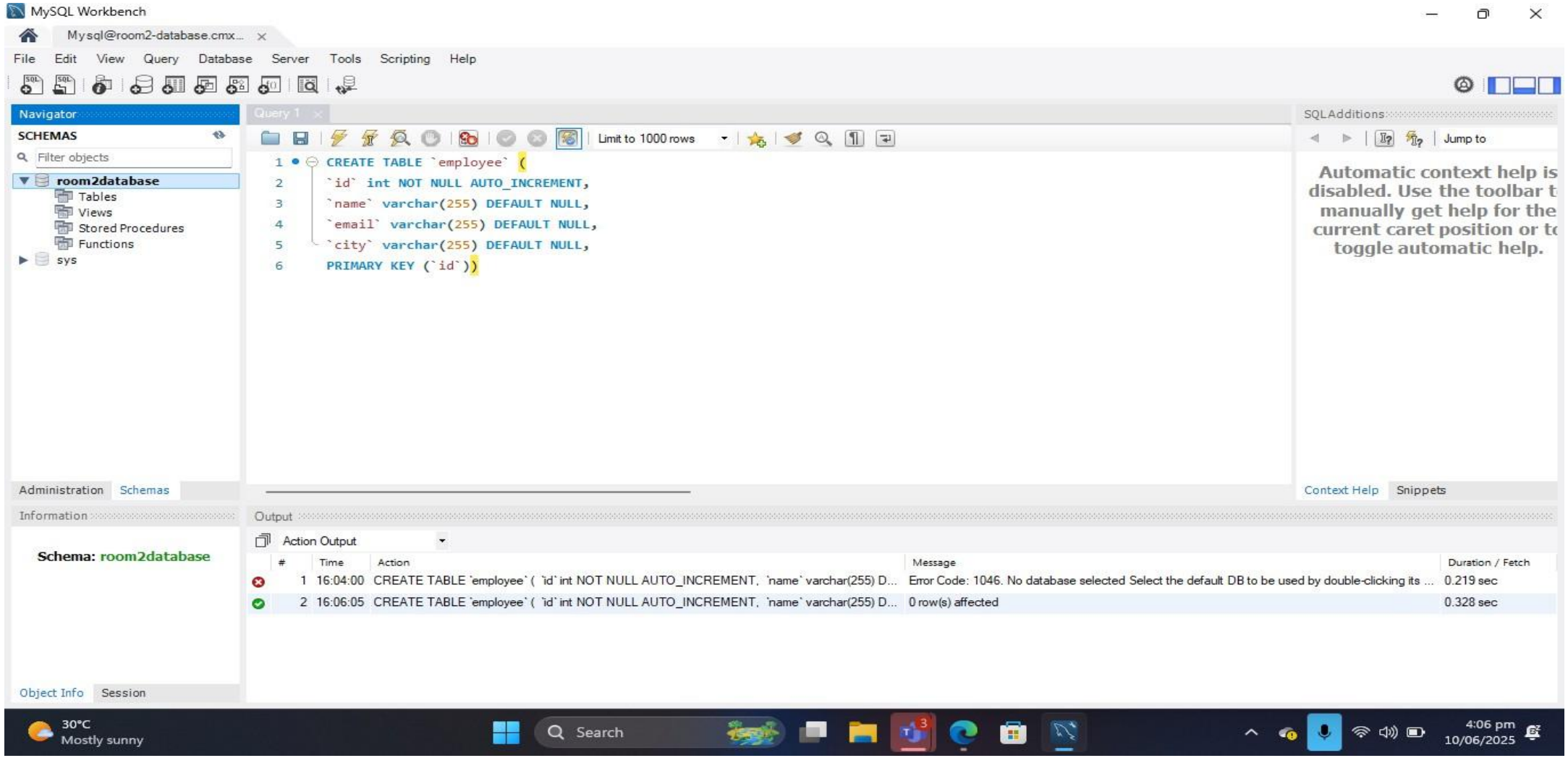
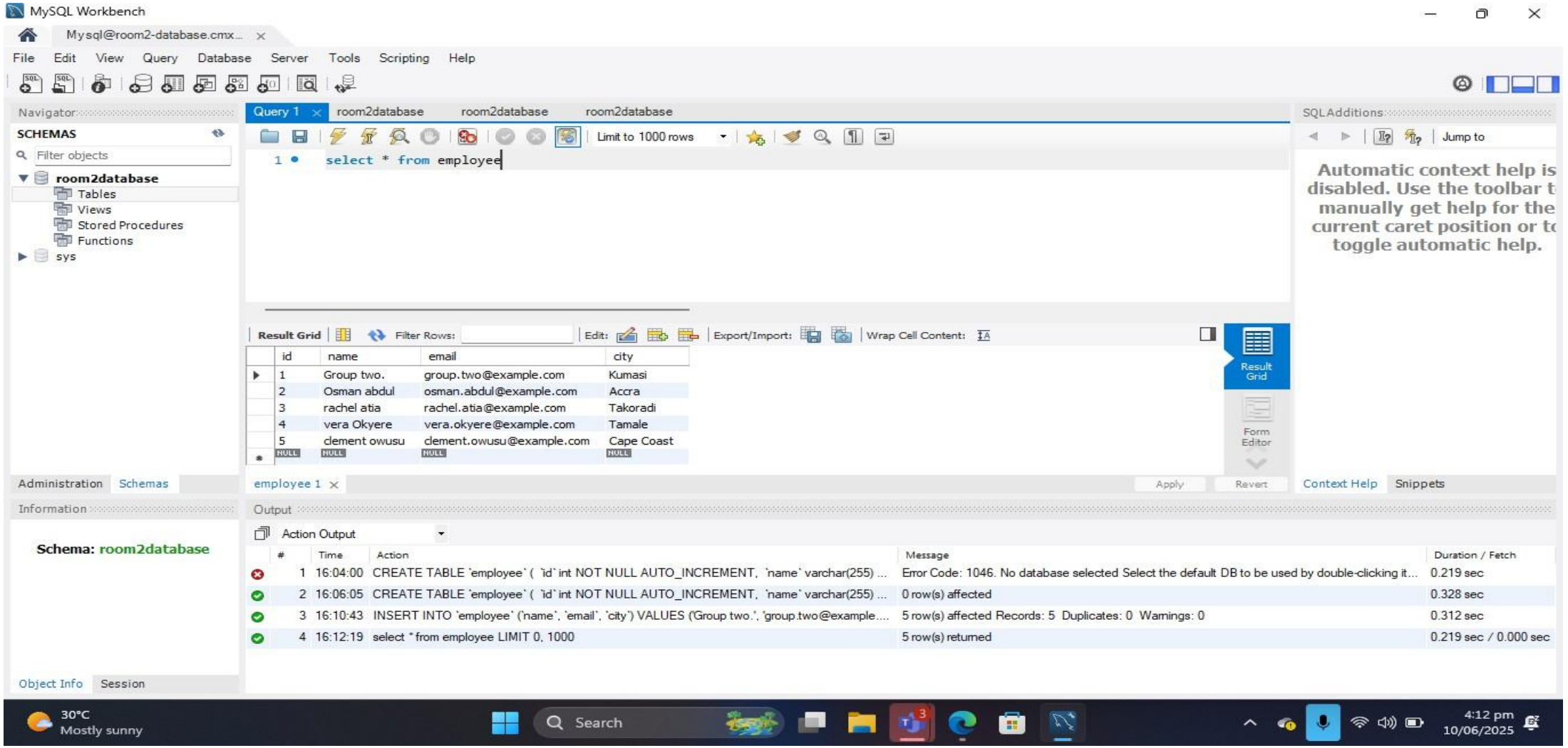


Fig. 3

Viewing of Data in "employee" table



Conclusion and Summary

This project provided hands-on experience in deploying a managed MySQL database using Amazon RDS, connecting to it securely, and executing SQL operations through MySQL Workbench. Throughout the process, key AWS concepts such as instance provisioning, networking, security groups, automated backups, and resource cleanup were explored.

Key Lessons Learned:

- AWS RDS simplifies database setup by abstracting infrastructure management, making it ideal for scalable applications and learning environments.
- Security group configuration is critical for enabling external access; public access must be combined with correct IP and port settings.
- MySQL Workbench is a powerful GUI tool for connecting to and managing cloud databases.
- Testing with basic SQL scripts validates that the database is fully operational.
- Always clean up unused resources (like DB instances) to avoid unnecessary charges and maintain a tidy cloud environment.

Final Thought:
Mastering the end-to-end workflow of creating, connecting to, and managing RDS databases lays a solid foundation for more advanced cloud-native data applications and AWS certification readiness.